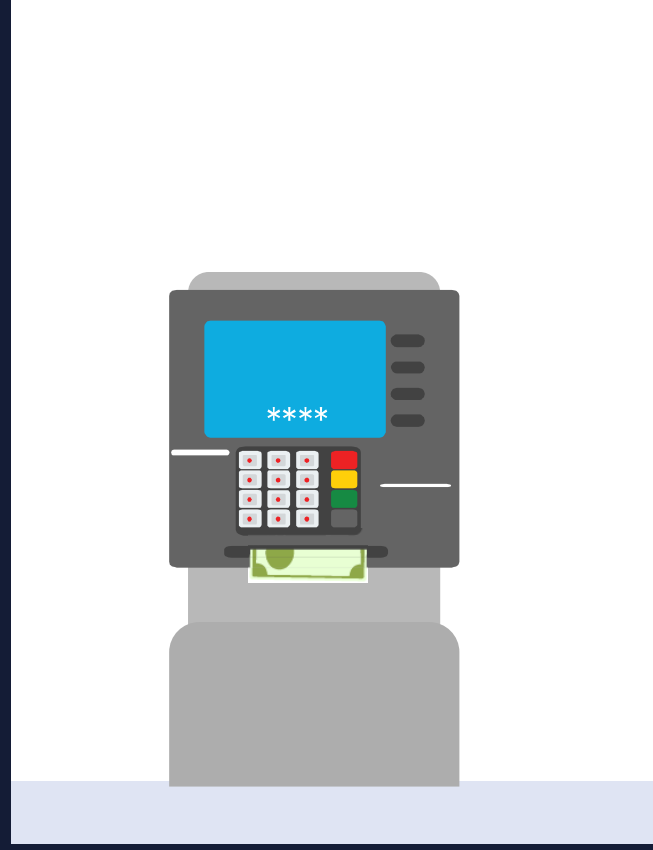
Get into Programming With JavaScript

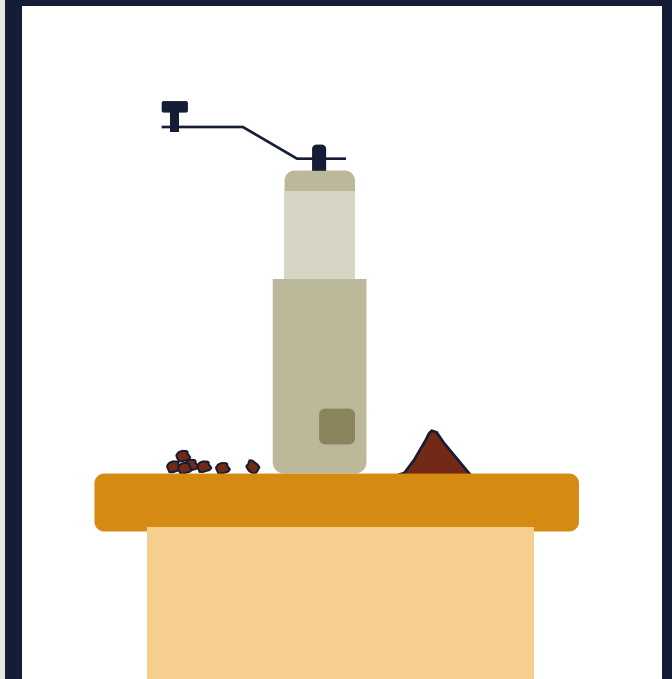
Axle Barr

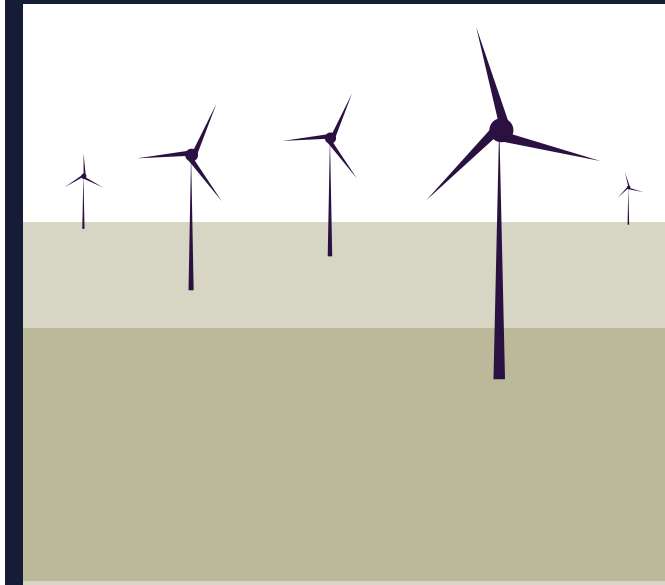
Repetition

In the real world, we:

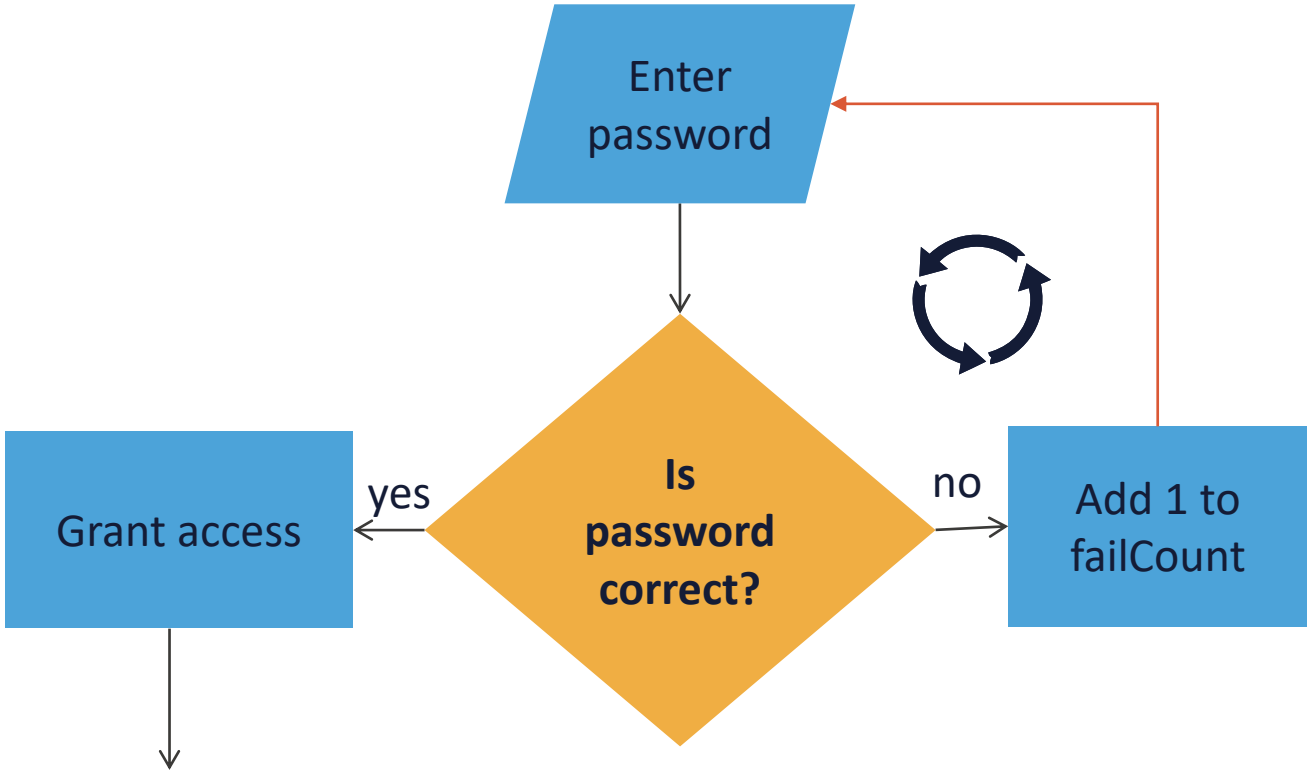
- Knock three times before someone answers the door
- Keep getting bills from the ATM util it completes the transaction
- Turn the coffee grinder until all the beans have been grounded
- See windmills turn until there is no wind







Flowchart for decisions



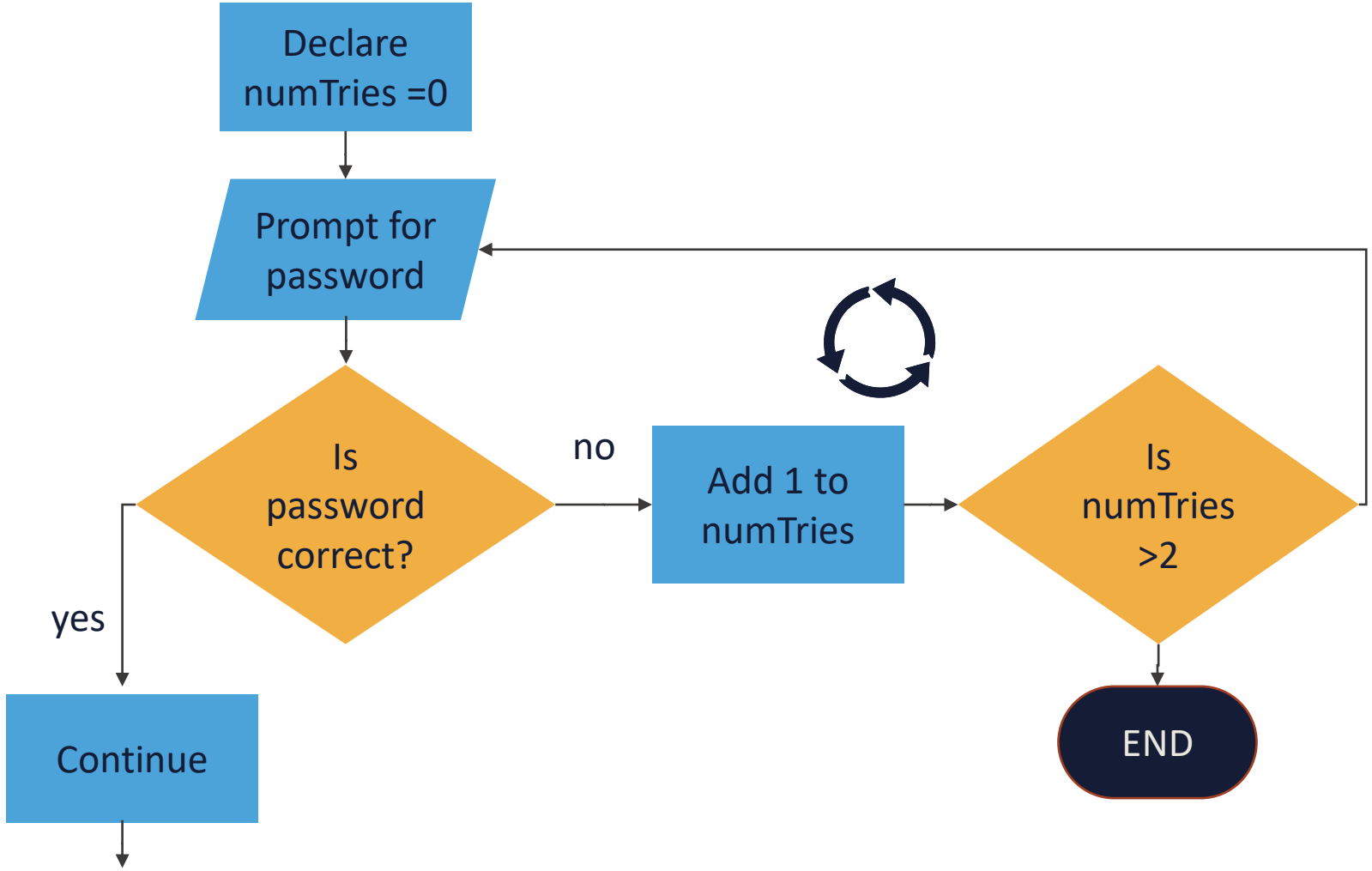
Diamonds are for Decisions

The diamond represents a decision or selection process. It represents some kind of change in the sequence of processing. There can only be two outcomes from a decision, a positive track or a negative track.

The diamond also helps us setup a loop.

We will discuss the switch statement later in the course.

Flowchart for entering password



If three fails, program ends

Algorithm

Steps performed by machine:

Prompt user for password
Store password in *psswd*
While *psswd* is incorrect
    prompt user for password
End while
Continue with program
…

- **While *psswd* is incorrect** is the same as saying, “if the user continues to insert the wrong password” then we will continue to prompt the user for the correct password

- The End while statement is just to show a block of code within a looping structure.

```
prompt user for password
store password in psswd
declare counter = 0
while counter <= 2
    check if psswd is correct
    if psswd is incorrect
        prompt user for password
        add 1 to counter
end while
continue with rest of program...
```

psswd is correct

1. prompt user for password
2. store password in *psswd*
3. declare *counter* = 0
4. while *counter* <= 2
5. check if *psswd* is correct
6. if *psswd* is incorrect
7. prompt user for password
8. add 1 to *counter*
9. end while
10. continue with rest of program…

psswd is NOT correct

1. prompt user for password
2. store password in *psswd*
3. declare *counter* = 0
4. while *counter* <= 2
5. check if *psswd* is correct
6. if *psswd* is incorrect
7. prompt user for password
8. add 1 to *counter*
9. end while
10. continue with rest of program…

psswd is NOT correct

More Operators

Operators already seen:

=

>

<

=>

<=

+ - * /

Two more operators

== Equality Operator

!= Inequality Operator

“Axle” == “Axle”

“Axle” == “Axl”

9 == 9

“Axle” != “Axel”

Shopping Cart Loop

Here is a simple program that:

1. Asks for the price of a product
2. Changes that string into a float data type
3. Calculates the final price after adding TAX
4. Prints the final price

```
const TAX = 1.08
let moreProducts = true;
while (moreProducts == true) {
        let productCost = prompt("Enter price of product: ");
        productCost = parseFloat(productCost);
        finalPrice = productCost * TAX;
        //moreProducts = false;
        let moreItems = confirm("Do you have more items?");
        if(moreItems == false){
                moreProducts = false;
        } else {
                moreProducts = true;
        }
}
printOut = finalPrice;
```

Shopping Cart Loop

Here is a simple program that:

1. We need to keep track of the total

```
const TAX = 1.08
let moreProducts = true, total=0.0;
while (moreProducts == true) {
        let productCost = prompt("Enter price of product: ");
        productCost = parseFloat(productCost);
        finalPrice = productCost * TAX;
        total = total + finalPrice;
        let moreItems = confirm("Do you have more items?");
        if(moreItems == false){
                moreProducts = false;
        } else {
                moreProducts = true;
        }
}
printOut = total;
```

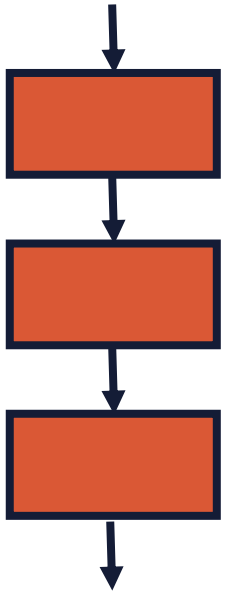
Shopping Cart Loop

Here is a simple program that:

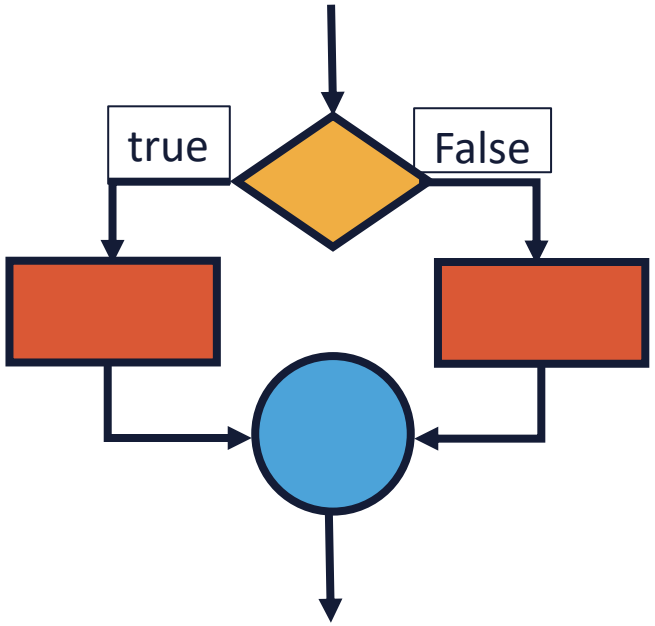
1. Total can now be printed

```
const TAX = 1.08
let moreProducts = true, total=0.0, productCost=0.0,
moreItems=false, productCost=0.0, finalPrice = 0;
while (moreProducts == true) {
        productCost = prompt("Enter price of product: ");
        productCost = parseFloat(productCost);
        finalPrice = productCost * TAX;
        total = total + finalPrice;
        moreItems = confirm("Do you have more items?");
        if(moreItems == false){
                moreProducts = false;
        } else {
                moreProducts = true;
        }
}
printOut = "Total: " + total;
```

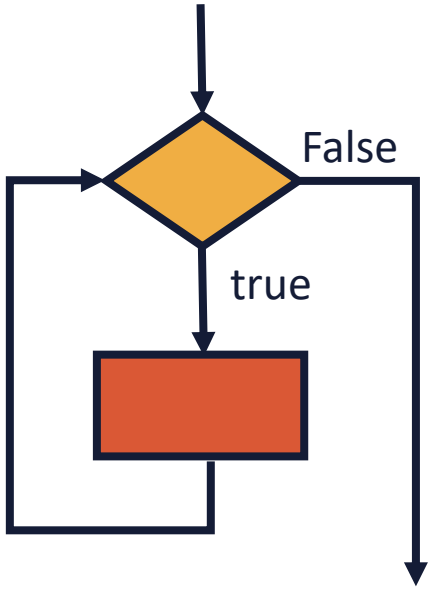
Only three programming structures



Sequence

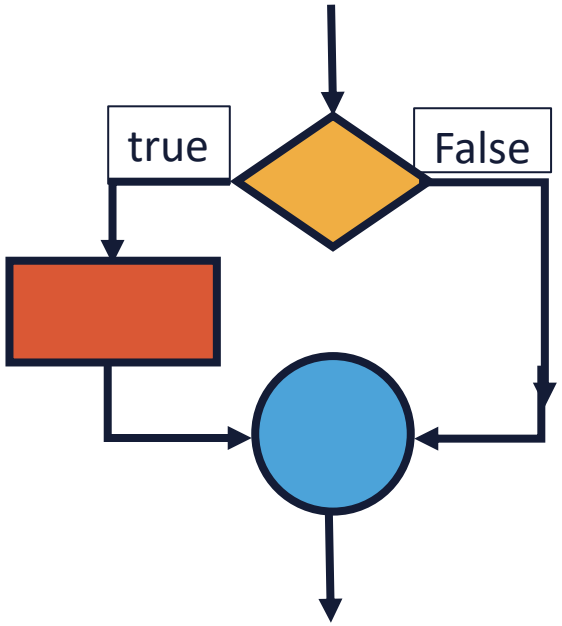


Decision/Selection



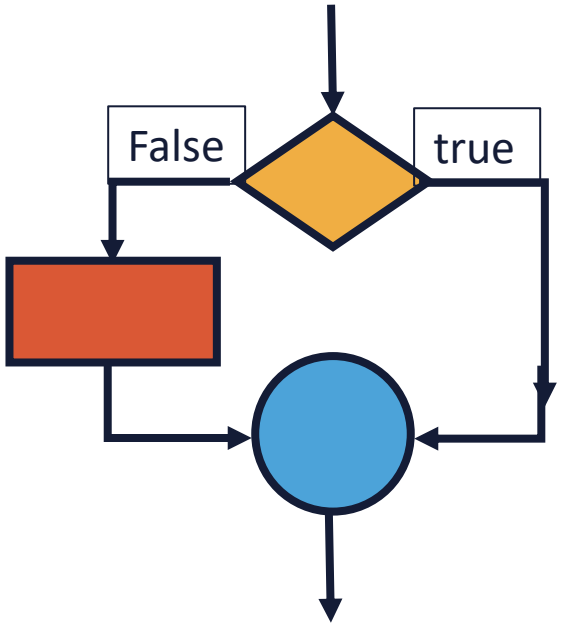
Iteration/Repetition

Only three programming structures



Decision/Selection

Only three programming structures



Decision/Selection

Shopping Cart

Main Program

```
const TAX = 1.08
let moreProducts = true, productCost=0.0,
finalPrice=0.0, total=0.0, moreItems=false;
while (moreProducts == true) {
  productCost = prompt("Enter price of product: ");
  productCost = parseFloat(productCost);
  finalPrice = productCost * TAX;
  total = total + finalPrice;
  moreItems = confirm("Do you have more items?");
   if(moreItems == false){
     moreProducts = false;
   } else {
     moreProducts = true;
   }
}
```

Shopping Cart

Main Program

```
const TAX = 1.08
let moreProducts = true, productCost=0.0,
finalPrice=0.0, total=0.0, moreItems=false;
while (moreProducts == true) {
 productCost = prompt("Enter price of product: ");
 productCost = parseFloat(productCost);
 finalPrice = productCost * TAX;
 total = total + finalPrice;
 moreItems = confirm("…more items?");
  if(moreItems == false){
   moreProducts = false;
  } else {
   moreProducts = true;
  }
}
```

Shopping Cart

Main Program

const TAX = 1.08	1.08
let moreProducts = true, productCost=0.0,	
finalPrice=0.0, total=0.0, moreItems=false;	
while (moreProducts == true) {	moreProducts = true
productCost = prompt("Enter price of product: ");	
productCost = parseFloat(productCost);	productCost = 100
finalPrice = productCost * TAX;	finalPrice = 108
total = total + finalPrice;	total = 108
moreItems = confirm(“…more items?");	moreItems = true
if(moreItems == false){	moreItems = true
moreProducts = false;	
} else {	
moreProducts = true;	moreProducts=true
} }	
printOut = "Total: " + total;	

Shopping Cart

Main Program

const TAX = 1.08	1.08	1.08
let moreProducts = true, productCost=0.0,		
finalPrice=0.0, total=0.0, moreItems=false;		
while (moreProducts == true) {	moreProducts = true	moreProducts = true
productCost = prompt("Enter price of product: ");		
productCost = parseFloat(productCost);	productCost = 100	productCost = 200
finalPrice = productCost * TAX;	finalPrice = 108	finalPrice = 216
total = total + finalPrice;	total = 108	total = 324
moreItems = confirm(“…more items?");	moreItems = true	moreItems = false
if(moreItems == false){	moreItems = true	moreItems = false
moreProducts = false;		moreProducts = false
} else {		
moreProducts = true;	moreProducts=true	
} }		
printOut = "Total: " + total;		Total: 324

Shopping Cart

Main Program

const TAX = 1.08	1.08	1.08	1.08
let moreProducts = true, productCost=0.0,			
finalPrice=0.0, total=0.0, moreItems=false;			
while (moreProducts == true) {	moreProducts = true	moreProducts = true	moreProducts = true
productCost = prompt("Enter price of product: ");			
productCost = parseFloat(productCost);	productCost = 100	productCost = 200	productCost = 300
finalPrice = productCost * TAX;	finalPrice = 108	finalPrice = 216	finalPrice = 324
total = total + finalPrice;	total = 108	total = 324	total = 648
moreItems = confirm("…more items?");	moreItems = true	moreItems = **true**	moreItems = false
if(moreItems == false){	moreItems = true	moreItems = **true**	moreItems = false
moreProducts = false;		moreProducts = **true**	moreProducts=false
} else {			
moreProducts = true;	moreProducts=true	moreProducts=true	
} }			
printOut = "Total: " + total;			Total = 648

JavaScript's *for...next* loop

Alternative to the while...end while loop

01

```
for (let i = 0; i < 6; i++) {
  printOut = "i: " + i;
}
```

02

```
for (let i = 0; i < 9; i++) {
  printOut = printOut + "i: " + i;
}
```

03

```
for (let i = 0; i < 6; i++) {
  printOut = printOut + "i: " + i + "<br />";
}
```

04

```
let total = 0;
for ( let i = 0; i < 4; i++ ) {
  total += I ;
}
```

JavaScript's *for...next* loop

Alternative to the while...end while loop

01

```
let total = 0, i = 0;
for (i; i < 4; i++) {
  total+=i;
}
printOut = "Total: " + total;
```

02

```
let total = 0, i = 0;
for (i; i < 9; i++) {
  total+=i;
  if(i==5) break;
}
```

03

```
let total = 0, i = 2;
for (i; i < 9; i++) {
  total+=i;
  if(i==5) break;
}
printOut = "Total: " + total;
```

JavaScript's *for...next* loop

Alternative to the while...end while loop

01

```
for let i=0; i < 6; i++) {
  for(j=0; j<5; j++){
    printOut += "*";
  }
  printOut += "<br />";
}
```

02

```
for let i=0; i < 6; i++) {
  for(j=i; j<5; j++){
    printOut += "*";
  }
  printOut += "<br />";
}
```

03

```
for (let i=6; i >= 0; i--) {
  for(let j=i; j<5; j++){
    printOut += "*";
  }
  printOut += "<br />";
}
```